Through the Looking Glass

Visualization of requirements and test analysis networks

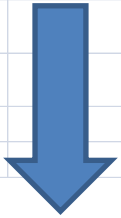
Sam Brown
L-3 Communications

Motivation for a Visualization Methodology

Studying characteristics of information flow in large Requirements sets

>40 documents
>20,000 tests or requirements
>25,000 linkages

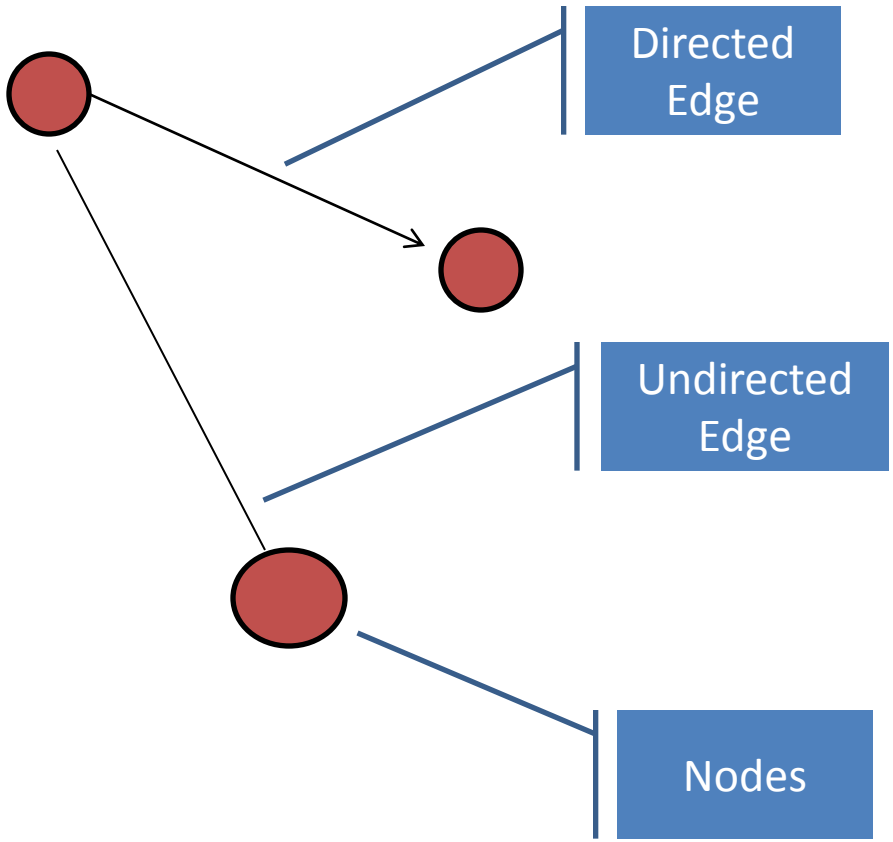
LMA	Requirement	FSW	Requirement	TLCM	Requirement
387	The flight system shall support the DOR tone capability in the SDST, including wideband DOR tones at X-band.	1317	The flight software shall command the transponder as defined by the transponder documentation	706	The flight software shall configure the transponder telemetry inputs in accordance with the active FS side whenever the trnsponder is powered ON. Reference transponder ICD for selection table.
1997	The flight system shall accommodate PCM / PSK / PM modulation for the X band downlink.			711	The flight software shall provide the capability to command an “active” telecom side which determines the “active” transponder in use and the uplink channel.
				712	The flight software shall propagate the power state and configuration of the transponder when changing the “active” telecom side.
				713	The flight software shall provide a default “active” telecom side upon initialization. The default “active” telecom side, in the absence of faults or obituary table entries, will be Side 1 (Telecom Side 1 uses SDST 1, and Telecom Side 2 uses SDST 2).
				714	The flight software shall only perform the necessary SDST initializations if a commanded “active” telecom side is different from the currently “active” telecom side.
				725	The flight software, upon initial application of transponder power ON, shall provide for a configurable default state. Subsequent power ON transitions will default to last commanded state.
				1200	The flight software shall provide the capability to enable or disable the X-Band exciter for the active transponder.
				1201	The flight software shall provide the capability to enable or disable c mode for the active transponder.
				1202	The flight software shall provide the capability to enable or disable X-Band Ranging for the active transponder.
				1203	The flight software shall provide the capability to set the Ranging Modulation Index for the active transponder.
				1204	The flight software shall provide the capability to enable or disable X-Band Differential One-Way Ranging (DOR) Mode for the active transponder.
				1205	The flight software shall provide the capability to command an X-Band convolutional encoding mode of TLM_OFF, rate 7 1/2, or BYPASS for the active transponder.
				1206	The flight software shall provide the capability to to command the Ranging Mode to BASEBAND or EXTERNAL for the active transponder.
				1207	The flight software shall provide the capability to command the X-Band Subcarrier for the active transponder to one of the following frequencies: 281.25 Khz squarewave, 281.25 Khz sinewave, 25 Khz squarewave, or 25Khz sinewave.
				1208	The flight software shall provide the capability to command the X-Band Squarewave Telemetry Modulation Index to one of 128 discrete values (0x00 to 0x7F) for the active SDST.
				1209	The flight software shall provide the capability to command the X-Band Sinewave Telemetry Modulation Index to one of 16 discrete values (0x0 to 0xF) for the active SDST.
				1211	The flight software shall provide the capability to command the X-Band telemetry modulation mode to SUBCARRIER or BPSK for the active transponder.



Quickly communicate regarding patterns involving hundreds or thousands of requirements

Graphs and Networks

A very brief review of terms



Old idea(Euler,1731) but still very useful

Commonly used in network analysis and visualization

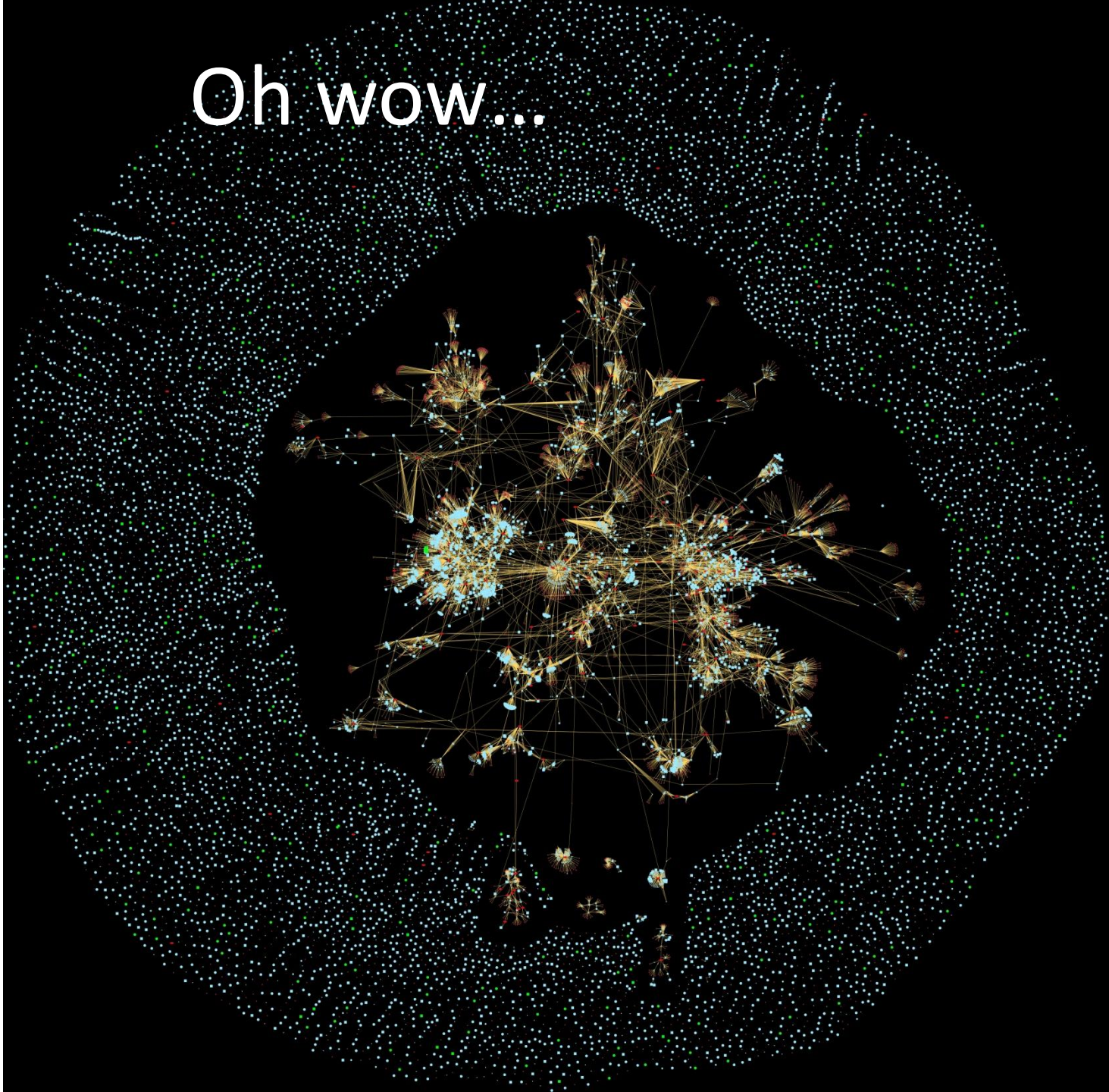
Core to network analysis and topology

Tools for Visualization

- GUESS
  - Standard and well-known graph tool
  - Freely available and very flexible
  - All examples in this presentation are from GUESS
- Extensions and programming
  - Gython/Jython/JavaSwing for search/queries
  - Quick way to write analysis scripts
  - JavaSwing allows for quick and elegant user interfaces
- ICD for DB to Guess/GRV translation
  - Simple way to generate GUESS datasets and record analysis results back into a managed database.
  - Supports most Access/RDB variable types
  - Provides for multi-user and configuration management
- New tools coming
  - Second generation of Guess by same developer (not yet available)
  - 3-D visualization (Walrus: too many limitations for this application)

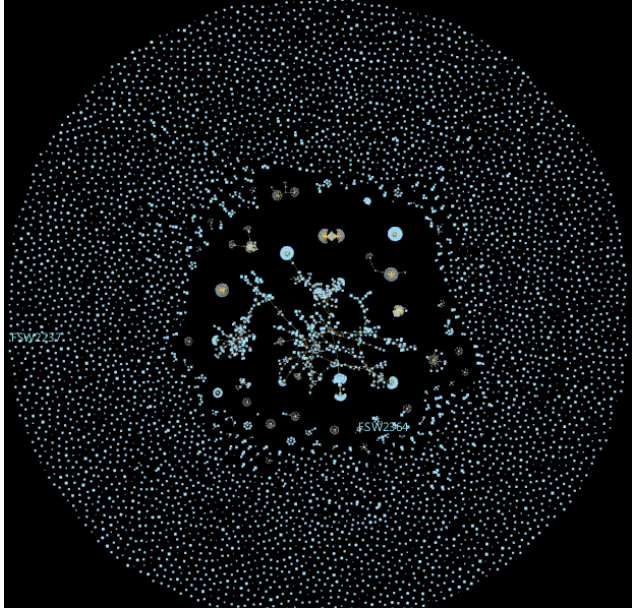
Oh wow…

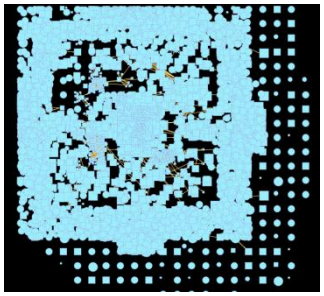
- Flight system requirements/ test network for operational S/C
- Dealing with large models requires care and some computer time
- This model has 20K nodes and 25K edges

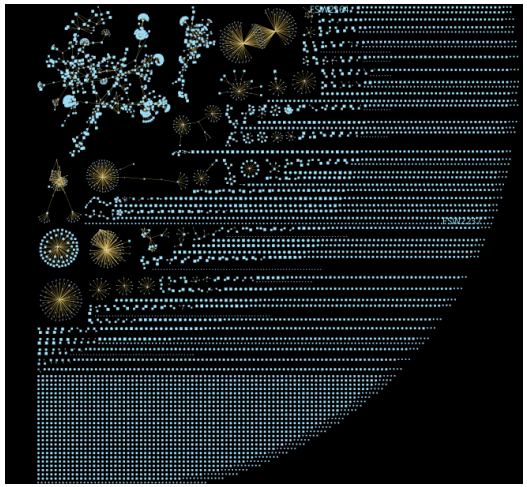


Layouts

- GEM
  - Reliable, usually useful, slow for large models
  - Most examples in this presentation are GEM layouts (Oh wow)
- Spring/physics
  - Poor results on the larger models.
  - May be better for dynamical models
- Circle/Radial
  - Not useful for requirement/test networks
- MDS
  - Fast but ugly
- BINPACK
  - Fast and useful in some cases, but still a bit ugly
  - Separates subnets which can improve clarity
- Research opportunities
  - Hierarchical algorithm taking advantage of known network structure
  - Fast but not yet ready for prime time.



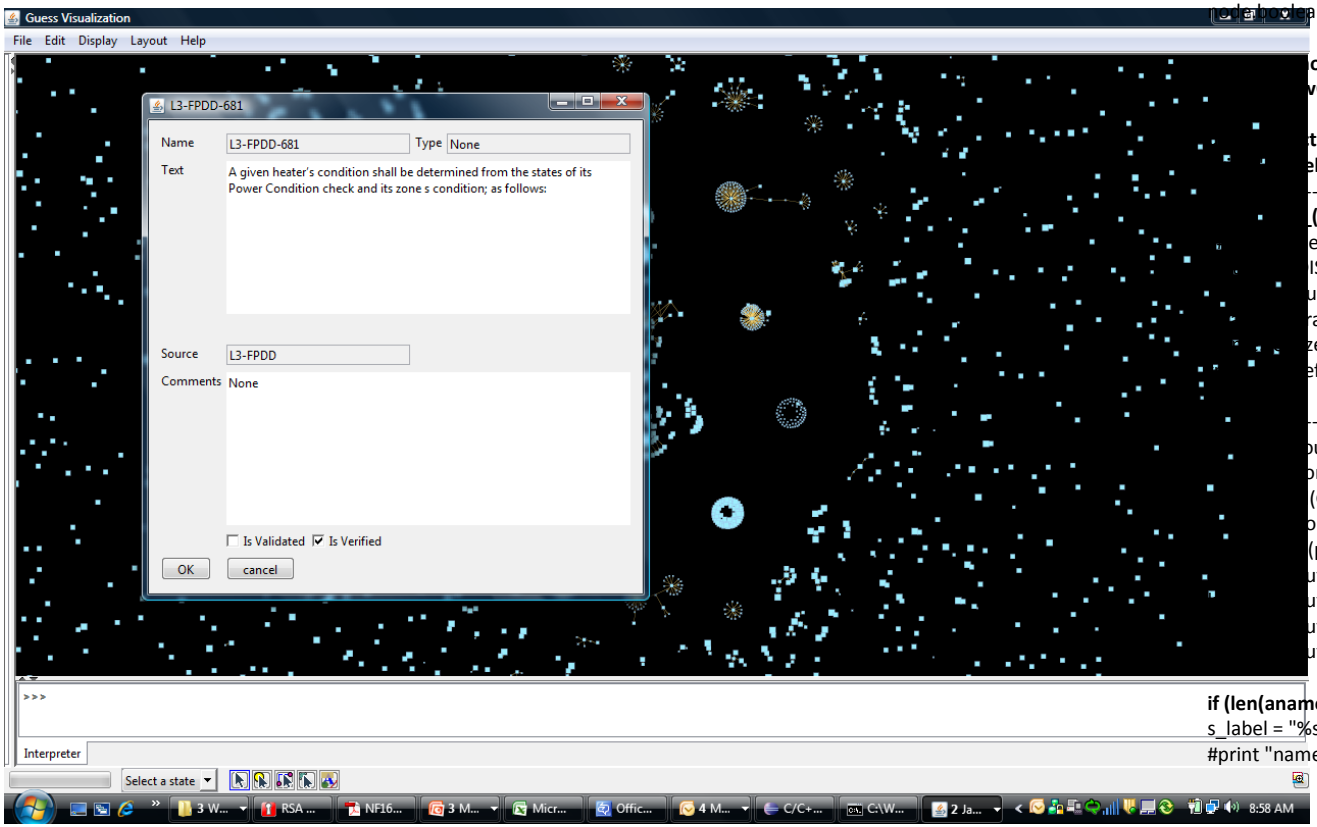




Editing/Exploring Using GRV

Simple jython script with rdb interface

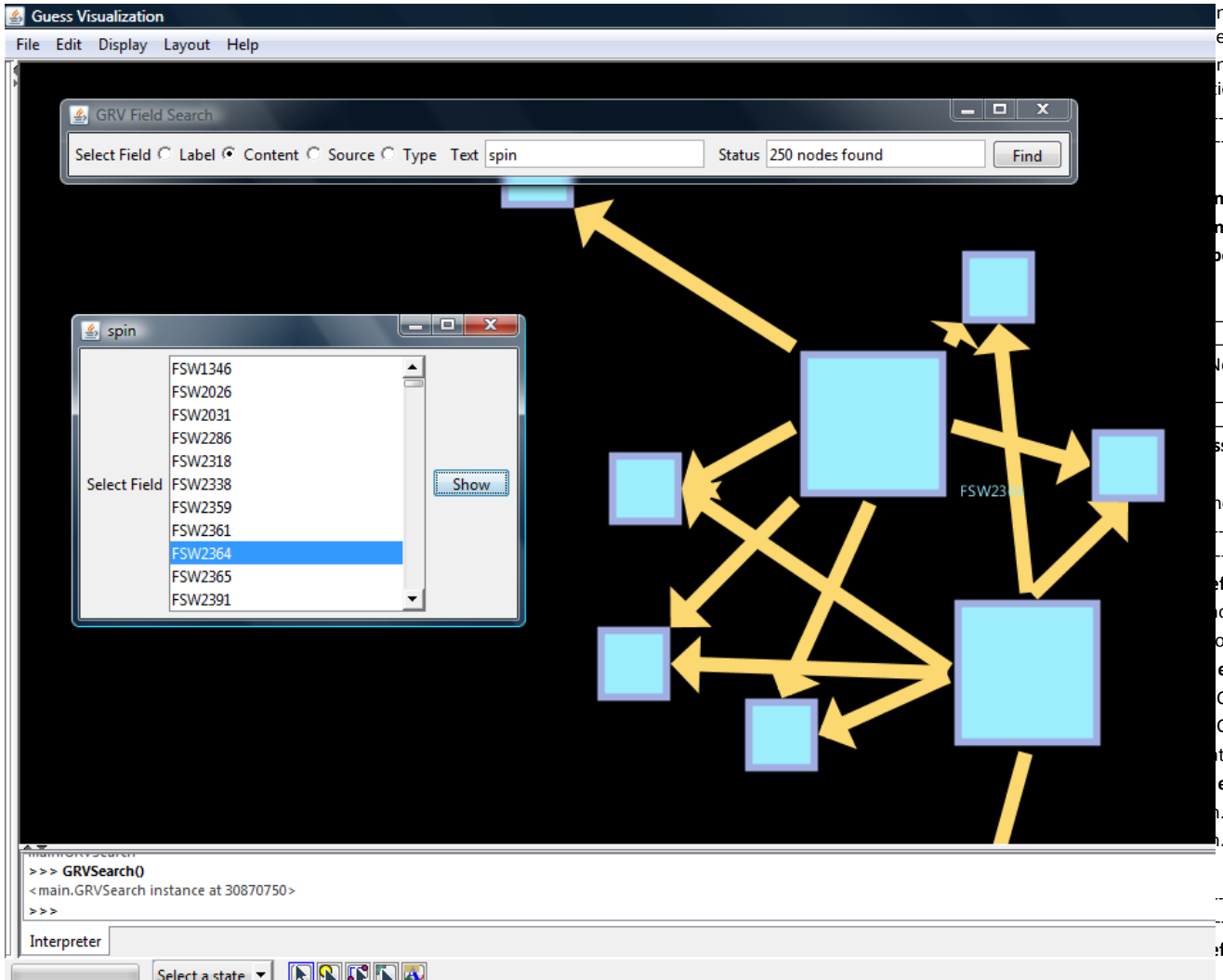
~150 sloc using swing java dialogs




```
# create NodeShow window
#_____________________________________________________________
# Make a search window to find strings within name, text, source or type
#_____________________________________________________________
class GRVNodeShow:
#  show name text type and source
#    editable comments IsValid IsVerified
#    keep running tab of face time
#  for the later edge show then
#shownode1, node2
#editable comments IsValid IsVerified
#keep tab of face time
# note that aname is a LIST of nodes and we have to select [0] to get the first
def NodeShow_OK(self,event):# this is the OK button event handler
node = (name == self.nametextfield.getText())
print "Selected =" , node.label
node.memo2 = self.ctextarea.getText()# copy changed fields into the record
node.boolean1 = self.cb_IsVerified.isSelected()
node.boolean2 = self.cb_IsValid.osSelected()# how do we close the window

ow_Cancel(self,event):# this is the cancel button event handler
vClosing()

t(self, event):
elect"
----------

(self, aname):
es setup the basic frame, size.
ISPOSE_ON_CLOSE lets it return back to Guess
u click the close button
rame(aname.label[0])
ze(500, 500)
efaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE)

---------- Create Panel and add swing components and
oupLayout(frame.getContentPane())
ontentPane().setLayout(layout)
(GroupLayout(frame))
out(layout)
(pnl)
utoCreateGaps(true);
utoCreateContainerGaps(true);
utoCreateGaps(true);
utoCreateContainerGaps(true);

if (len(aname) >= 1):# prep for error - cannot be dups
s_label = "%s" %aname.label[0]
#print "name = "+ s_name
```

Searching Using GRV

Swing Java/Jython script
~145 sloc



- #Swing JTextField for Juno Guess.
- #----------
- #General purpose Juno field search.
- # Centers view on node and neighbors
- # Displays node label
- #----------
- #srb - NASA IVV
- #Mar 2011
- # components

no_Search - finds a list of nodes matching search
eria
no_ShowNode - selects node to display with
ions

---

m javax.swing import *
m java.lang import *
ort time

Node selector - show function uses list from search

---

ss GRVShow:

how a node from the list - responds to ShowButton

---

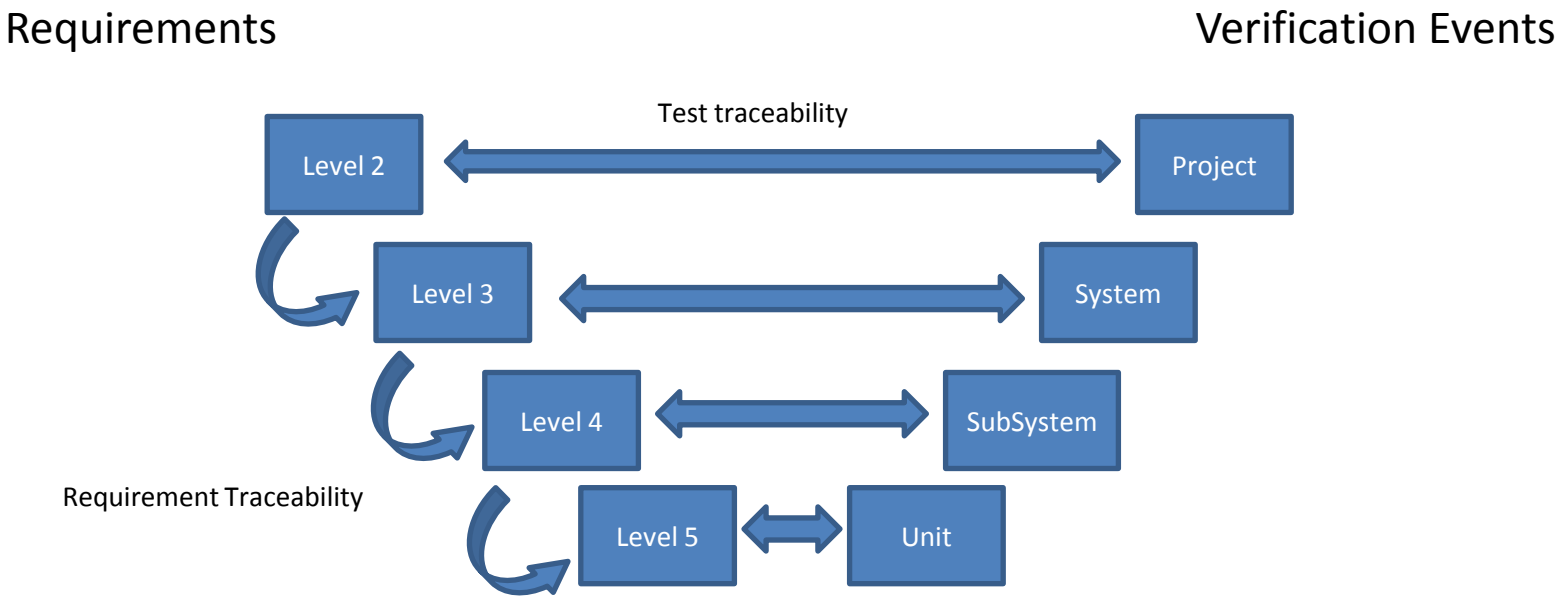
f showit(self, nodelist):
ode = (label == **self.list.getSelectedValue())**
oCenter = []
elm in anode:
Center += elm.getNeighbors()
Center += anode
ter(_toCenter)
elm in anode:
.labelvisible = 1
.labelcolor = yellow

---

f __init__(self, nodelist, pattern):

Outward Signs of Internal Troubles

- Patterns associated with difficulties
  - Hourglass (multiple inheritance)



Traceability is key to both requirements development and requirements verification
Each project has unique approaches to traceability and verification

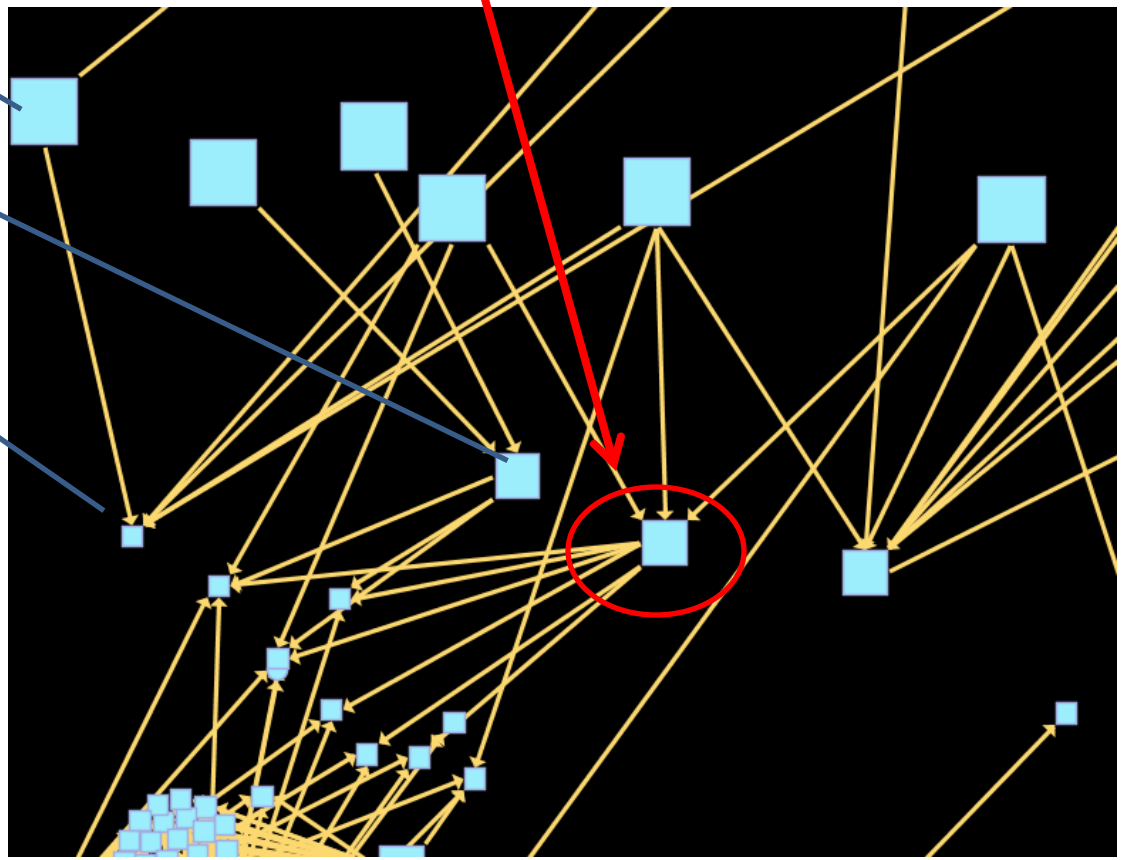
Pitfalls of Multi-Parenting

Three parents/Six children
Direct links to children from unrelated grandparents

L3 Reqmts

L4 Reqmts

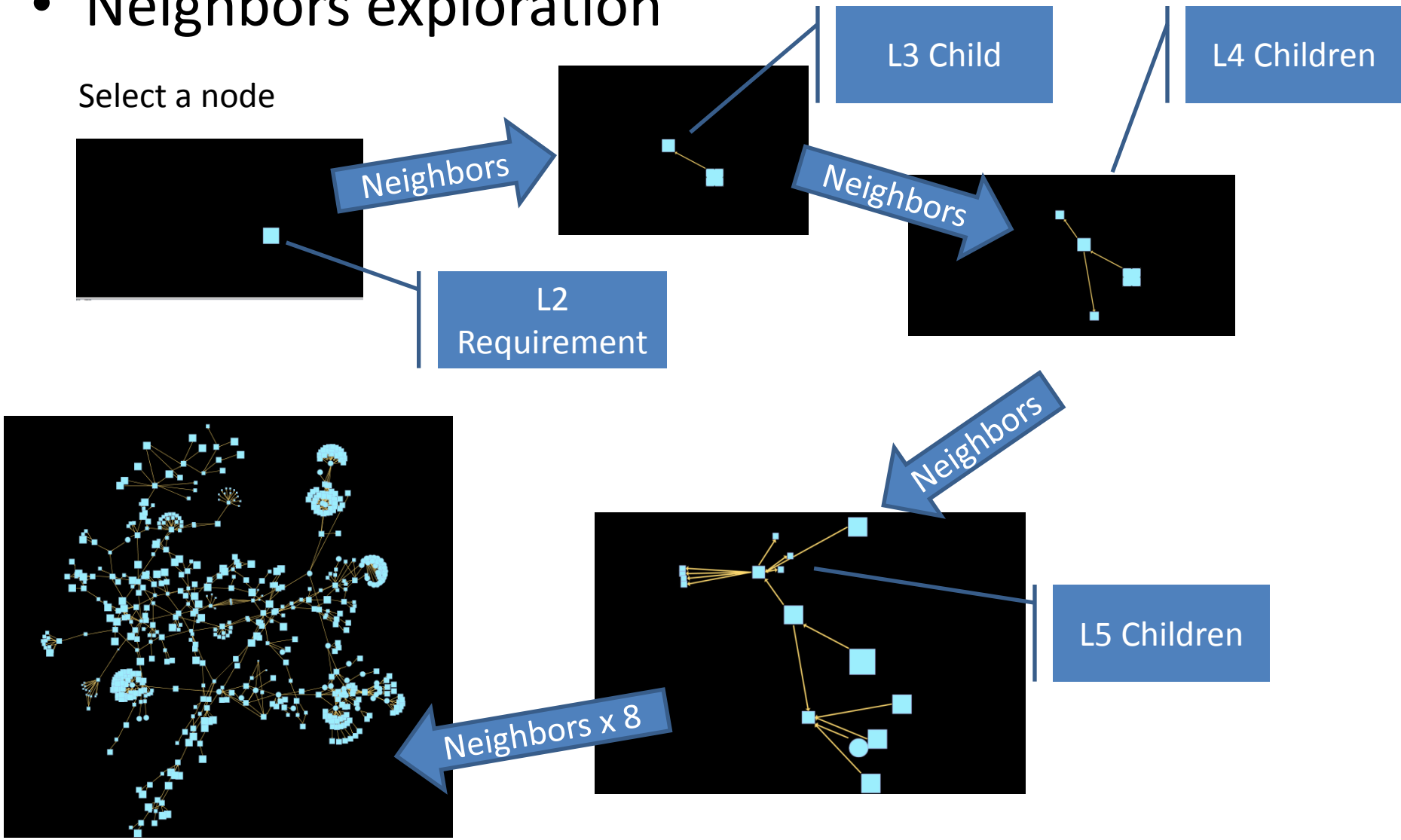
L5 Reqmts



Validation difficulties
Testing considerations
Time-phasing of development

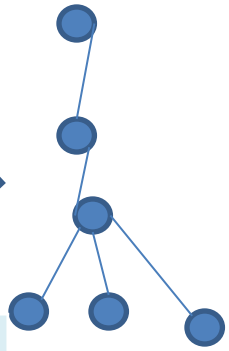
Connectivity Studies

- Neighbors exploration



The cutting edge

- Connection statistics to support mathematical approaches
  - Methods for determinants of 20K by 20K connection matrices (sparse) – Fiedler number det(D-A)
  - Validation of methodology across multiple projects



A Matrix

0	1				
1	0	1			
	1	0	1	1	1
		1	0		
		1		0	
		1			0

Laplacian (D-A)

1	-1				
-1	2	-1			
	-1	4	-1	-1	-1
		-1	1		
		-1		1	
		-1			1

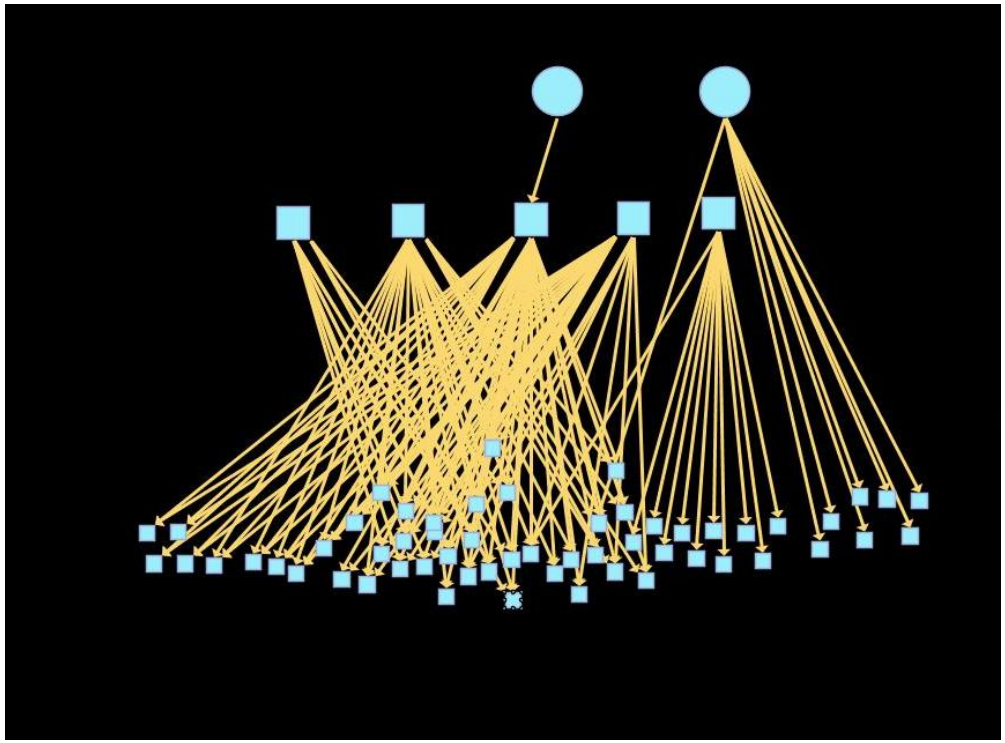
Eigenvalue:
From linear algebra
$Lx=\lambda x$ where λ is an eigenvalue
And x is a non-null eigenvector
Because L is symmetric the eigenvalues are all real

$$\lambda=\{0, 0.486, 1, 1, 2.428, 5.086\}$$

Fiedler number = 0.486
implying somewhere between an expander (1) and a tree form (1/6)

Stuff that is not yet done

- Hierarchical layout script
  - Group Req by document <- easy
  - Position based on level <- easy
  - Sort to minimize edge crossings <- hard!



And Stuff that is Almost Done

- Fully automatic bidirectional database/graphing tools
  - ICD a first step (third version)
  - Core jython scripts for GUESS are working well
  - Minor development to complete an Access/RDB interface

Summary

- Graphical approaches can be useful tools alongside traditional methods
- It is possible to visualize large models and quickly draw meaningful conclusions
- Jython/swing java tools are fast enough for even a very large models (20K nodes)
- Plenty left to explore for the imaginative.